



Kauno technologijos universitetas

Informatikos fakultetas

Registrai

P175B100 Skaitmeninės logikos pradmenų trečias laboratorinis darbas

Projekto autorius

Gustas Klevinskas

Akademinei grupei

IFF-8/7

Vadovai

Doc. Tomas Adomkus

Kaunas, 2019

Turinys

Įvadas	3
Universalus registras	3
Teisingumo lentelė.....	3
Rezultatai	3
Specializuotas registras	4
Teisingumo lentelė.....	4
Rezultatai	5
Perėjimas prie FPGA.....	6
Išvados	8

Įvadas

Darbo tikslas – susipažinti su įvairių tipų registrais, jų struktūra, veikimu, taikymo galimybėmis ir realizavimu naudojant trigerius. Išsiaiškinti postūmių operacijas ir jų atlikimo būdus. Patikrinti jų veikimą programuojamos logikos schemoje.

Užduotys:

1. Apsirašyti registro veikimo lentelę pagal duotas sąlygas;
2. Realizuoti registrą;
3. Parašyti stimulą ir patikrinti suprojektuoto registro funkcionavimą;
4. Pakoreguoti schemą ir ją įkelti į FPGA plokštę.

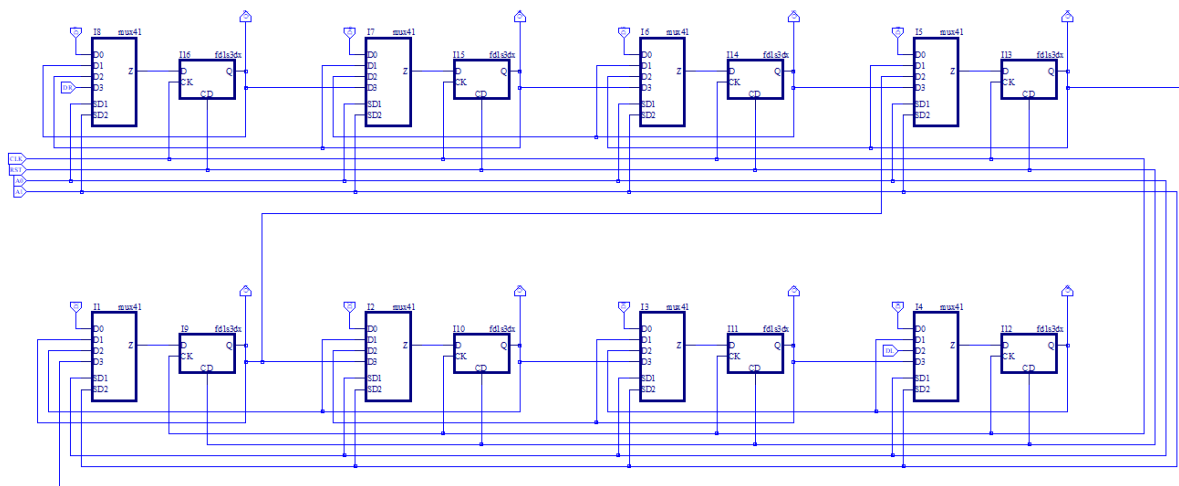
Universalus registras

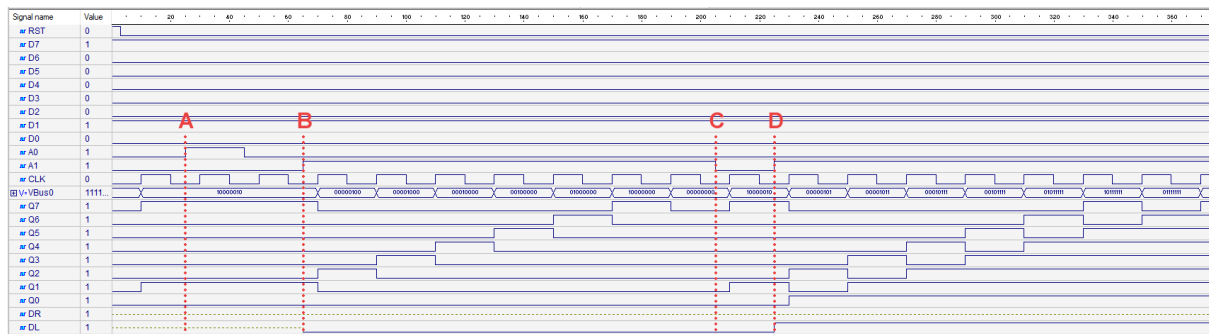
Teisingumo lentelė

A ₀	A ₁	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Paiškinimas
0	0	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Įrašymas
1	0	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Saugojimas
0	1	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	D _L	LL, D _L
1	1	D _R	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	LR, D _R

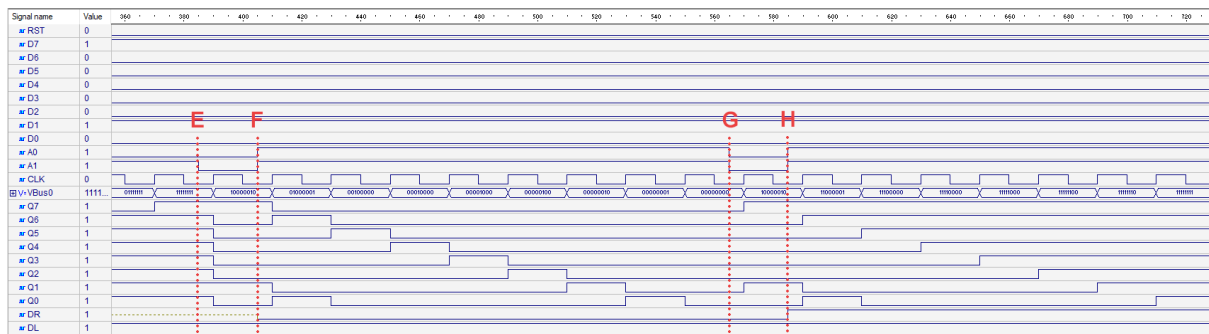
Lentelė 1. Universalaus registro teisingumo lentelė.

Rezultatai





Pav. 2. Universalus registro laiko diagrama 1.



Pav. 3. Universalus registro laiko diagrama 2.

Taške A – pereinama į saugojimo režimą. Matome, kad išlieka prieš tai įrašyme įrašyta informacija.

Taške B – pradeda vykti loginis postūmis į kairę įrašant 0. Gale registras užsipildo nuliais.

Taške D – pradeda vykti loginis postūmis į kairę įrašant 1. Gale registras užsipildo vienetais.

Taške F – pradeda vykti loginis postūmis į dešinę įrašant 0. Gale registras užsipildo nuliais.

Taške H – pradeda vykti loginis postūmis į dešinę įrašant 1. Gale registras užsipildo vienetais.

Taškuose C, E ir G – iš naujo įrašoma informacija į registrą.

Specializuotas registras

Teisingumo lentelė

Duotoje užduotyje reikėjo sukurti specializuotą registrą:

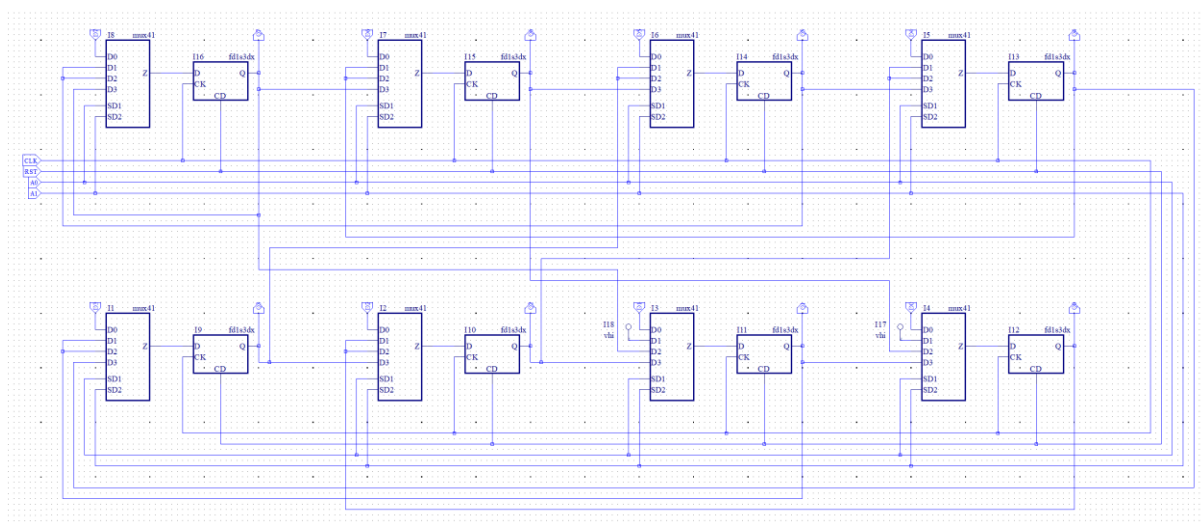
- 8 bitų;
- atlieka 3 postūmius:
 - loginį į kairę per 2, įrašant 1;
 - ciklinį į kairę per 2;
 - aritmetinį į dešinę per 1, papildomu kodu;
- nulis nustatomas asinchroniškai.

Žinant reikalavimus galima sudaryti registro veikimo lentelę. Ji padės jungiant signalus prie multiplexerių įvesčių.

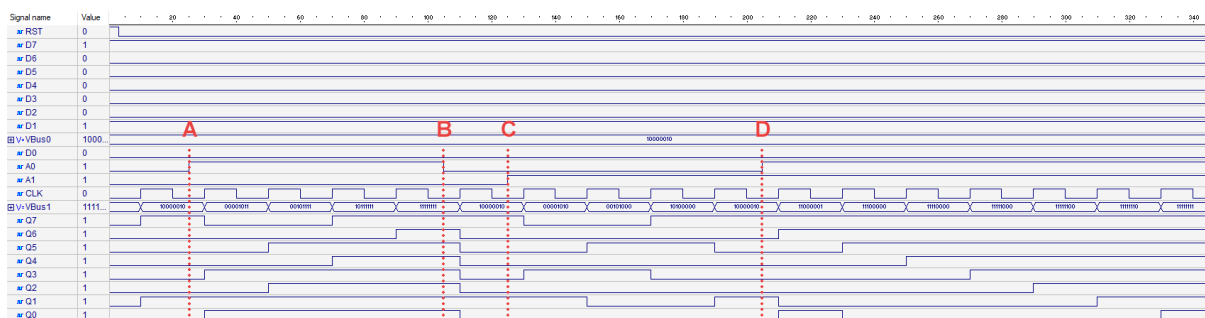
A ₀	A ₁	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Paaškinimas
0	0	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	Įrašymas
1	0	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	1	1	LL ₂ , 1
0	1	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	Q ₇	Q ₆	CL ₂
1	1	Q ₇	Q ₇	Q ₆	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	AR ₁ , P

Lentelė 2. Specializuoto registro teisingumo lentelė

Rezultatai



Pav. 4. Specializuoto registro schema.



Pav. 5. Specializuoto registro laiko diagrama.

Kad būtų paprasčiau matyti postūmius, registro išvestis sudėjau į VBus1. Laiko diagramoje sužymėjau svarbius laiko momentus.

Taške A – A₀ ir A₁ reikšmės pereina iš 0, 0 į 1, 0. Tai atitinka loginį postūmį. Matome, kaip ties kylančiu sinchrosignalo (CLK) frontu registras pradeda vykdyti šį postūmį ir juos tęsia atėjus naujam kylančiam sinchrosignalo frontui. Postūmiai tęsimi kol registras užsipildo vienetais.

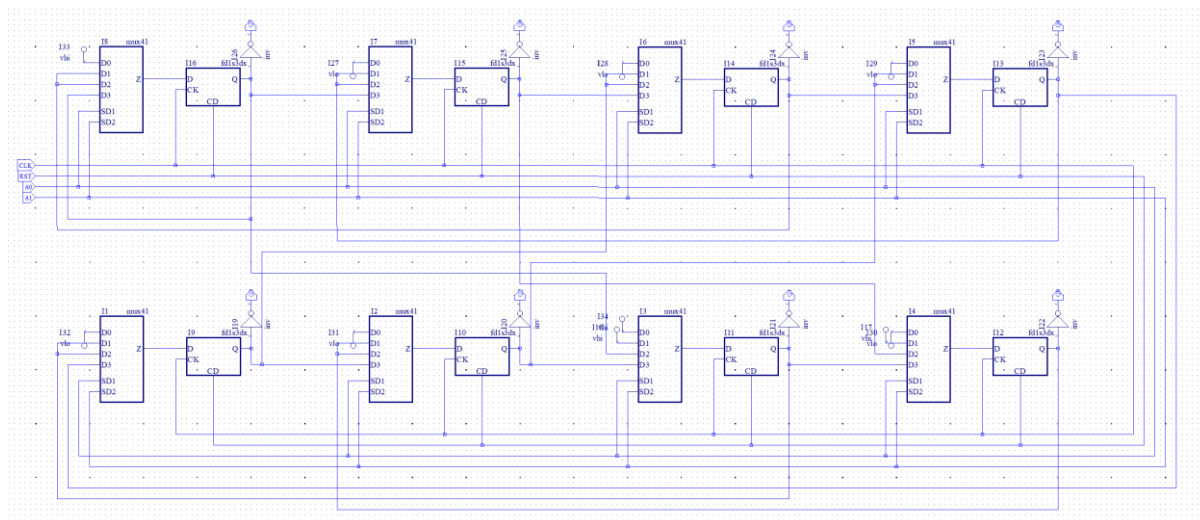
Taške B – A₀ ir A₁ reikšmės grąžinamos į 0, 0 tam, kad grąžintume pradinis duomenis į registrą (nes prieš tai buvo užpildytas vienetais). Tai įvyksta atėjus kylančiam CLK signalui.

Taške C – A₀ ir A₁ reikšmės pereina į 0, 1. Tai atitinka ciklinį postūmį. Matome, kad praėjus keturiems CLK signalo periodams registro vertės grįžta į pradines.

Taške D – A₀ ir A₁ reikšmės pereina į 1, 1. Tai atitinka aritmetinį postūmį į dešinę. Kadangi registre naudojamas papildomas kodas, atsilaisvinusiose vietose įrašoma ženklo vertė (šiuo atveju 1). Tai tęsiama, kol registras užsipildo vienetais.

Perėjimas prie FPGA

Tam, kad šią schemą galėtume įkelti į FPGA plokštę, ją reikės truputį pakoreguoti. Pirmiausia pakeičiau D signalus (vertes, kurios įkeliamos į registrą) kad jos būtų „kietai“ įrašytos, t. y. prie jų prijungiau *vlo* ir *vhi* signalus. Po to reikėjo invertuoti registro išvestis, nes FPGA plokštėje LED užsidega, kai signalas būna 0 (arba teoriškai galėtume pakeisti LED režimą į „pull-down“).



Pav. 6. FPGA plokštei pritaikyta registro schema.

Po to priskiriame registro įvestis ir išvestis konkrečioms fiziniams kontaktams per „Spreadsheet View“ įrankį.

	Name	Group By	Pin	BANK	VREF	IO_TYPE	PULLMODE
1	All Ports	N/A	N/A	N/A	N/A		
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	A0	N/A	58(58)	5(5)	N/A	LVC MOS2...	UP(UP)
1.1.2	A1	N/A	57(57)	5(5)	N/A	LVC MOS2...	UP(UP)
1.1.3	Clock	N/A	N/A	N/A	N/A	N/A	N/A
1.1.3.1	CLK	N/A	50(50)	5(5)	N/A	LVC MOS2...	UP(UP)
1.1.4	RST	N/A	55(55)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	Q0	N/A	46(46)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.2	Q1	N/A	45(45)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.3	Q2	N/A	44(44)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.4	Q3	N/A	43(43)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.5	Q4	N/A	40(40)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.6	Q5	N/A	39(39)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.7	Q6	N/A	38(38)	5(5)	N/A	LVC MOS2...	UP(UP)
1.2.8	Q7	N/A	37(37)	5(5)	N/A	LVC MOS2...	UP(UP)

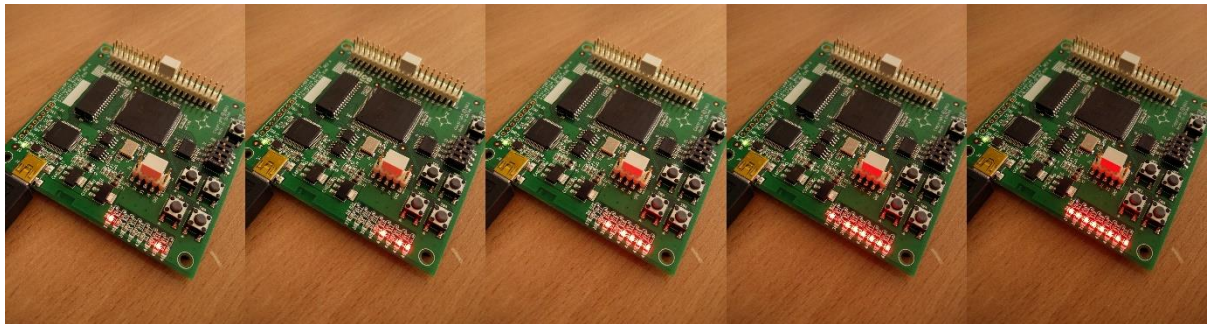
Pav. 7. „Spreadsheet View“ langas.

Belieka sugeneruoti kodą ir užprogramuoti FPGA plokštę. Pateiksiu nuotraukas, kuriose matosi plokštės veikimas.

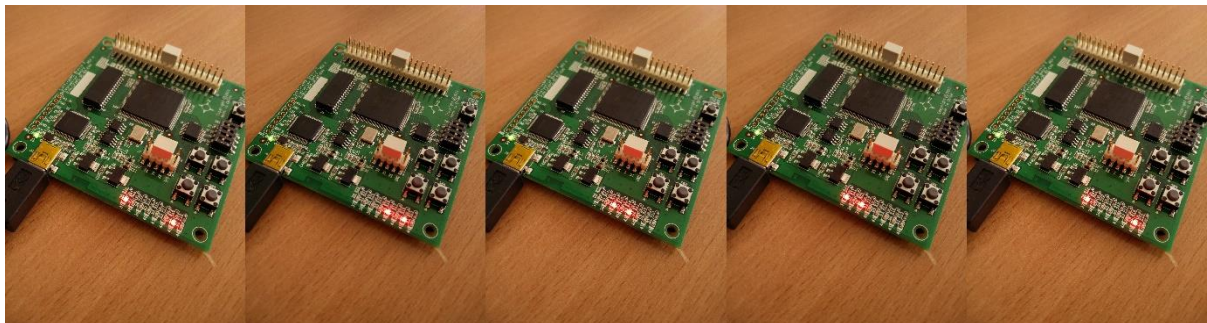
5 paveikslėlyje apibrėžiau keturis jungiklius. Du iš jų atitinka multiplekserio valdymo signalus, ir vienas „reset“ komandą (jei „reset“ signalas būtų ant mygtuko, jį reikėtų visad laikyti nuspaustą, nes mygtukas yra „pull-up“ režime. Teoriškai jei pakeistume mygtuko režimą į „pull-down“, jis veiktų teisingai, tačiau man pakeitus režimą jis vis tiek veikė „pull-up“ režimu. Spėju, kad šioje plokštėje fiziškai nėra „pull-down“ funkcijos).



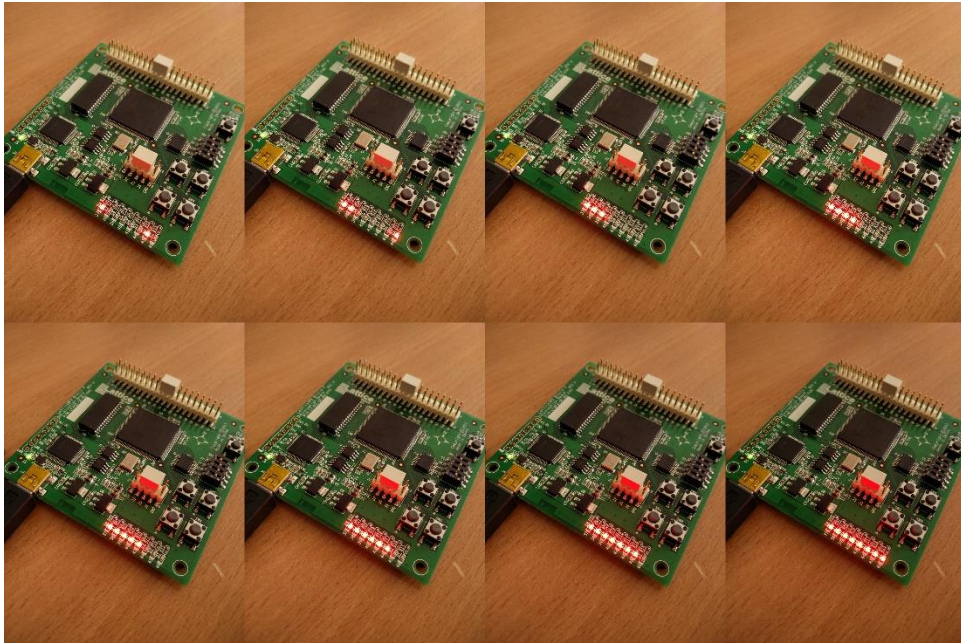
Pav. 8. Plokštė duomenų įrašymo režime.



Pav. 9. Loginis postūmis į kairę per 2 vietas įrašant 1.



Pav. 10. Ciklinis postūmis į kairę per 2 vietas.



Pav. 11. Aritmetinis postūmis papildomu kodu į dešinę per 1 vietą.

Išvados

Darbas buvo atliktas sėkmingai; testuojant registrą jis grąžino reikiamus rezultatus.

Kadangi registras buvo sudarytas iš D trigerių, jis atlikdavo postūmius vis atėjus naujam CLK signalui. Tai buvo galima pamatyti iš laiko diagramos ir perkėlus schemą į FPGA plokštę – kiekvieną kartą paspaudus CLK mygtuką įvykdavo poslinkis.